

OpenGlyph

Named, font-backed historical Chinese glyphs

OpenGlyph contributors
Maintainer: Zejia FAN

Version 0.2.1 — 28 October 2026
CTAN submission candidate

Abstract

OpenGlyph lets an author insert a rare glyph by its ASCII identifier and typeset it as a real font character. An optional offline compiler constructs new forms from component templates. The built-in collection contains 22,583 GlyphWiki `ebag_s` forms with 38,462 source and dependency programs. This package is a typesetting tool, not an OCR system or a scholarly authority. It neither determines a character's reading nor proves that a generated form matches a manuscript. Compare glyphs with the source before publication.

Contents

1	Start with an existing form	2
2	Engines, installation and dependencies	2
2.1	Personal TEXMF installation	2
2.2	Project-local and Overleaf use	2
3	LaTeX interface	3
3.1	Low-level declarations	3
4	Construct a new form offline	4
4.1	A small, complete project	4
4.2	Editing vocabulary	5
5	Reusable modules and selected records	5
6	Stable identity and exchange	6
7	Rebuilding, licensing and review	7

1 Start with an existing form

After installation, this is a complete document:

```
\documentclass{article}
\usepackage{openglyph}
\OpenGlyphLoad{ebag}
\begin{document}
An existing form: \ebag{s030-067}.
\end{document}
```

The output includes 𐄂. Further source IDs produce 𐄃 and 𐄄. These are font glyphs, not images. The command `\ebag{s030-067}` abbreviates `\OpenGlyph{ebag:s030-067}`. You need not enter a rare Chinese character or memorize its private-use codepoint.

The offline HTML index, `openglyph-index.html`, searches source IDs and component references. A component hit is only a visual search lead, not a claim about character identity. The index uses the supplied OTF; it does not request glyph images from a server.

2 Engines, installation and dependencies

Use XeLaTeX or LuaLaTeX with LaTeX2e dated 2020-10-01 or newer. The runtime depends on `fontspec`, `iftex` and `xparse`. pdfLaTeX is not supported. No shell escape or network service is used during ordinary typesetting. Tectonic can use the project-local layout described below. The Chinese example additionally uses `ctex` and `Fandol`, and `LuaTeX-ja` when compiled with LuaLaTeX. This English manual uses `geometry`, `booktabs`, `array`, `longtable`, `fancyvrb` and `hyperref`.

2.1 Personal TEXMF installation

Find your personal tree with `kpsewhich -var-value=TEXMFHOME`. Within that tree install:

```
tex/latex/openglyph/openglyph.sty
tex/latex/openglyph/openglyph-ebag.def
fonts/opentype/public/openglyph/OpenGlyphEbag.otf
doc/latex/openglyph/          (contents of doc/)
source/latex/openglyph/       (contents of source/)
```

Put the root README, license, notices and disclosure with the documentation. Preserve the internal directory layout of the documentation and builder. Refresh the distribution's filename database if necessary. The two files under `tex/` are the only runtime TeX files. Example files named `catalog.tex` must not be installed globally as runtime files.

The optional builder stays in the source tree and is invoked explicitly by Python. No executable wrapper or shell-escape permission is needed. A TeX installation does not supply its Python or Node.js dependencies.

2.2 Project-local and Overleaf use

Alternatively place `openglyph.sty`, `openglyph-ebag.def` and `OpenGlyphEbag.otf` beside the paper's main source. The original local layout `ebag/catalog.tex` with its font beside it remains

supported. A local **ebag/** bundle takes precedence over the installed catalogue. Use only one version of each glyph namespace in a document. For custom bundles, specify an explicit relative or absolute directory path and keep the font there. Discovery solely through an unrelated `TEXINPUTS` search path is not supported.

Copy the example `examples/myglyphs/` directory alongside the two example sources to compile them locally. An Overleaf project only needs prebuilt fonts and catalogues; font construction is a separate offline step.

3 LaTeX interface

Command	Purpose
<code>\OpenGlyphLoad{ebag}</code>	Load the built-in collection; repeated loads of the same argument do nothing.
<code>\OpenGlyphLoad{myglyphs}</code>	Load a local font bundle. Its catalogue is <code>myglyphs/catalog.tex</code> ; the font stays alongside the catalogue.
<code>\OpenGlyph{ebag:s030-067}</code>	Print an explicitly named glyph.
<code>\ebag{s030-067}</code>	Short form for a built-in glyph.
<code>\gwglyph{my:n2}</code>	Synonym for <code>\OpenGlyph</code> ; often used for custom forms.
<code>\OpenGlyphAlias{short}{my:n2}</code>	Add a document-specific alias to a previously loaded glyph; aliases must be unique.
<code>\OpenGlyphSequence{my:n1,my:n2}</code>	Print a comma-separated list of IDs in the specified order.
<code>\OpenGlyphIfExistsTF{id}{yes}{no}</code>	Branch on whether the ID is registered.

Glyphs follow the current font size. They can occur in running text, footnotes¹ and headings. They are boxed individually to isolate CJK inter-character font switches. The supplied font has one drawing style; bold or italic variants are not automatically designed. Whole-document vertical writing and critical-edition layouts require other packages and are not provided here.

Unknown IDs, duplicate definitions, missing font files and missing character mappings produce errors. Compile with `-halt-on-error`; do not accept a PDF produced by continuing past errors. Some nonfatal interactive errors display a bracketed identifier to make failure visible. The package does not silently substitute a visually similar common character. With `hyperref`, bookmarks use bracketed ASCII IDs instead of assuming that a PDF bookmark can carry a custom font.

3.1 Low-level declarations

These declarations define a font and map an ID to a hexadecimal codepoint:

```
\OpenGlyphDeclareFont{namespace}{filename.otf}
\OpenGlyphDeclare{id}{namespace}{E000}
```

Generated catalogues use the faster `\OpenGlyphDeclareEntry` interface after Python validation. This is an implementation detail for generated, trusted catalogues, not a general sanitizer for

¹A font-backed footnote: 醫.

untrusted TeX. IDs begin with an ASCII letter and then use letters, digits, colon, period, underscore or hyphen.

4 Construct a new form offline

Install Python 3.10+, Node.js, FontTools and skia-pathops. In the `source/` directory:

```
python3 -m venv .venv
. .venv/bin/activate
python -m pip install -r requirements-build.txt
python scripts/openglyph.py inspect ebag:s030-067 \
    --node r4 -o preview
python scripts/openglyph.py build examples/newforms.json \
    -o myglyphs
```

On Windows, use `.venv\Scripts\Activate.ps1` for activation. Existing output folders are never overwritten. Open `preview/index.html` to see actual component occurrences and their path IDs. Rendering an isolated component may change stroke-avoidance details; also inspect the complete form.

Copy the generated `myglyphs/` into the paper project and add `\OpenGlyphLoad{myglyphs}`. Then use `\wglyph{my:n2}`.

4.1 A small, complete project

```
{
  "schema": "openglyph.project.v1",
  "namespace": "my",
  "glyphs": [{
    "id": "n2",
    "base": "ebag:s030-067",
    "edits": [{
      "op": "insert", "parent": "root", "id": "inner_dot",
      "glyph": "primitive:dot", "box": [77,129,119,167]
    }]
  }]
}
```

The compiler derives filled vector outlines through KAGE, creates an OpenType/CFF font, and writes a matching catalogue, allocation table, manifest and exact source programs. Normal paper compilation only reads these outputs. JSON input does not execute Python or raw JavaScript.

Operation	Base	New form
Replace one occurrence	𪛗	𪛗
Insert a dot template	𪛗	𪛗
Drop a record	𪛗	𪛗
Place a component	𪛗	𪛗

Each example independently edits the same base, rather than accumulating all four changes. These are synthetic forms without asserted readings or origins.

4.2 Editing vocabulary

Node paths such as `r4/r0` identify component occurrences, not every instance of a shared Glyph-Wiki part. Get the paths from `inspect`. For the example above, the four edits are:

```
{
  "op": "replace", "node": "r4/r0", "glyph": "u9f9c-j"
}, {
  "op": "insert", "parent": "root", "id": "inner_dot",
  "glyph": "primitive:dot", "box": [77, 129, 119, 167]
}, {
  "op": "drop", "node": "r4/r0/r2"
}, {
  "op": "place", "node": "r4/r0", "box": [28, 12, 92, 112]
}
```

`move` translates an occurrence with `delta:[dx,dy]`. `group` groups adjacent records in their existing order using `parent`, `id` and `nodes`. Inserted/grouped record IDs use an `a_` prefix. Existing record IDs are not renumbered after deletion. Local boxes use integer coordinates in the parent's 200-unit design space, not page-image pixels. The compiler preserves source program versions and transformations; it does not optimize a glyph against a scanned image.

5 Reusable modules and selected records

Version 0.2 introduces named modules. A module is itself a base plus edits; another edit refers to it as `module:marked_loop`. The supplied `examples/modules.json` contains:

```
{
  "schema": "openglyph.project.v1",
  "namespace": "modules",
  "glyphs": [{
    "id": "newform", "base": "ebag:s030-067",
    "modules": {
      "marked_loop": {
        "schema": "guwenzi.template-program.v1",
        "base": "primitive:loop",
        "edits": [{
          "op": "insert", "parent": "root", "id": "mark",
```

```

        "glyph": "primitive:dot", "box": [70,70,130,130]
    }
}
},
"edits": [{"op": "replace", "node": "r4/r0",
            "glyph": "module:marked_loop"}]
}]
}

```

Compile it with:

```
python scripts/openglyph.py build examples/modules.json \
    -o moduleglyphs
```

Load the resulting bundle and print `\gwglyph{modules:newform}`. Definitions are checked, including unused circular references. Nested `modules` dictionaries are not accepted; modules refer to definitions in the same dictionary.

An optional `select` takes existing records out of one parent:

```
"select": {"parent": "r4", "nodes": ["r0"]}
```

The paths and record names must come from inspection of the chosen base. Selection occurs before edits, so inspect the source and understand the new root before applying further edits. Records must remain in source order; unsupported scope-transform selections are rejected. This is structural reuse, not automatic decomposition of any arbitrary image.

6 Stable identity and exchange

Keep these concepts separate:

1. Source IDs name entries in a collection.
2. Occurrence paths name nodes in a particular template.
3. Project aliases identify an author's named glyphs.
4. A PUA codepoint addresses a glyph only within a font.
5. Hashes distinguish exact programs and outlines.
6. Scholarly identification and reading require research.

For revisions, preserve the old allocation file:

```
python scripts/openglyph.py build revision.json -o revised \
    --allocation myglyphs/allocation.json
```

Existing addresses stay fixed; retired addresses are not assigned to another ID. This does not mean a revised glyph has the same content hash. Always share the matching font, catalogue and manifest. The built-in collection uses standard Private Use Area A slots, not the old SEAL plane 10–13 scheme.

PDF readers need no separately installed font because fonts are embedded. Extracted text still contains font-local PUA characters: copying those into another font can produce a box or an unrelated form. The package does not promise semantic searchability or interoperable Unicode identities for new forms. Keep the source IDs and scholarly annotations with the editable work.

7 Rebuilding, licensing and review

The optional source tree includes the full selected GlyphWiki programs, address table, font-construction scripts and eight unmodified KAGE JavaScript files. From `source/`, run:

```
python scripts/build_ebag.py -o rebuilt
```

This reconstructs the built-in collection in a fresh folder.

Keep the locked address file, `data/ebag-addresses.json`. Do not assign new addresses from sort order when updating the collection. Rebuilding the full font is a separate, potentially lengthy operation, not part of document compilation.

OpenGlyph’s own code and documentation use MIT. GlyphWiki data and derived fonts retain the permissive upstream data grant; they are not relicensed as MIT or OFL. The optional KAGE engine is GPL v3, with corresponding source. See the root license, third-party notices and the license files under `source/`. No local SEAL font, private scan or unverified local IDS annotation dataset is included.

This work includes substantial AI assistance. See `AI-ASSISTANCE.md`. Automated tests do not replace knowledgeable human review, licensing review or scholarly glyph verification. This candidate does not certify completion of human review. It has not been submitted to or accepted by CTAN, TeX Live or MiKTeX.

Repository and issue tracker: <https://github.com/Open-Glyph/OpenGlyph-LaTeX>.